



Forum: Moteur de jeu GameBlender et alternatives

Topic: Programmation python dans le BGE

Subject: Re: Programmation python dans le BGE

Posté par: DaWaaaaghBabal

Contribution le : 6/2/2018 1:02:34

Il ne faut pas faire le moins de lignes possibles. Il faut écrire le code le plus clair et lisible possible. Ça passe, effectivement, par un code concis, mais typiquement quand tu es face à une méthode qui fait vingt lignes, la solution est pas d'enlever du code, c'est de le redécouper. La concision est un moyen d'atteindre la lisibilité, pas une fin en soi.

Quant à l'assignation, elle se fait même si la valeur est déjà la même. Le processeur est peut-être plus intelligent que ça, mais ça dépend du matériel, pas de Python, ne te pose donc pas la question.

Pour savoir si il peut intéressant de passer cette étape : non. Dans le cas actuel, tu fais *parfois* une assignation pour rien. En ajoutant un test pour vérifier si la valeur est la même ou pas, tu feras *systématiquement* un test, et *parfois* une assignation. L'assignation étant une opération élémentaire, ça ne vaut pas le coup.

Par contre, c'est de l'optimisation prématurée : première étape, un code qui marche ; deuxième étape, un code propre ; troisième étape, un code performant. Tant que tu n'as pas fini le module, tu n'en es pas à te demander si tu ne vas pas gagner une nanoseconde ici ou là.

Ensuite, on optimise d'abord les goulots d'étranglement, les points qui posent de réels problèmes de performance. Cf la loi de Pareto : 80% du temps de calcul se fait sur 20% du code, autant se concentrer sur ça, pas sur une assignation une fois par frame. Ce qui ne dispense pas de coder intelligemment, de bien structurer le code ou d'utiliser des algorithmes performants.

Pour aller plus loin... Ici, tu te poses la mauvaise question : ce n'est pas l'assignation qui pose problème, c'est l'initialisation de l'array [1, 0.2, 0, 1], c'est-à-dire une allocation et quatre assignations. Niveau performances, l'impact est toujours négligeable, mais considérablement plus élevé que l'assignation à la couleur. Et surtout, ce n'est pas propre, et ce pour deux raisons :

- On initialise le même tableau en boucle, ce n'est pas logique, la façon dont le code est écrit ne correspond pas à la façon dont il est *pensé* : tu veux "la couleur du soleil", pas "une nouvelle couleur du soleil".
- C'est un nombre magique. Cette valeur [1, 0.2, 0, 1] sort de nulle part, sans explication, et est codée en dur.

Bref, la couleur devrait être lue dans une propriété de l'objet, ou dans un fichier de configuration voire dans une constante globale quelque part, mais certainement pas initialisée à la volée et codée en dur.